自组织多目标粒子群优化算法 *

梁 静[1]，郭倩倩[1]，岳彩通[1]，瞿博阳[2]

(1. 郑州大学 电气工程学院，郑州 450001; 2. 中原工学院 电子信息学院，郑州 450007)

摘 要：针对多目标粒子群优化算法收敛性和多样性难以平衡的问题，提出一种利用问题的结构信息来解决多目标问题的自组织多目标粒子群算法。通过自组织映射网络发现种群和非支配解集分布的结构，构造出当前粒子的邻域关系，从邻域中选出非支配解，从而引导种群局部和全局的搜索。提出了精英学习策略，通过对精英粒子进行变异，引导算法跳出局部最优。实验结果表明，所提算法可以兼顾收敛性和多样性，有效地解决多目标优化问题。

关键词：多目标粒子群优化；自组织映射；种群分布；精英学习策略

中图分类号：TP183　　doi: 10.3969/j.issn.1001-3695.2018.01.0070

Self-organizing multi-objective particle swarm optimization algorithm

Liang Jing[1], Guo Qianqian[1], Yue Caitong[1], Qu Boyang[2]

(1. *School of Electrical Engineering*, *Zhengzhou University*, *Zhengzhou* 450001, *China*; 2. *School of Electric & Information Engineering*, *Zhongyuan University of Technology*, *Zhengzhou* 450007, *China*)

Abstract: In order to keep the balance between convergence and diversity of multi-objective particle swarm optimization algorithm, the paper proposed a self-organizing multi-objective particle swarm algorithm (SMPSO) , which utilized structure of population to solve multi-objective optimization problems. The distribution of the population and non-dominated solutions founded by self-organizing map network helped to construct the particle neighborhood relations , so as to improve the algorithm's local and global search ability by selecting non-dominated solution as leader in the neighborhood. The elite learning strategy could help population jump out local optimum by doing mutation on elite particles. The experimental results show its ability to keep convergence and diversity, and show its effectiveness to solve the multi-objective optimization problems.

Key words: multi-objective particle swarm optimization; self-organizing map; population distribution; elite learning strategy

0 引言

在现实应用中，有很多问题需要同时优化多个目标，两个及以上目标的问题叫做多目标优化问题(multi-objective optimization problems，MOP)。这些目标之间往往是相互冲突的，一个目标的性能提高会导致其他目标性能的降低，因此同时优化多个目标是很困难的。不同于单目标优化问题，多目标问题得到的不是一个解，而是由许多非支配解组成的解集，称为帕累托解集。帕累托解集在目标空间对应的目标值便组成了帕累托前沿。由于多目标问题得到的解可能有无限多个，所以找出所有的解是不实际的。在实际操作中，找出一定数量的非支配解，尽可能覆盖并靠近真正的帕累托前沿，且这些解分布均匀，是本文所需要解决的挑战性难题。

解决多目标问题的方法有两大类，一类是传统的数学解析性算法，一类是进化算法。进化算法是一类以种群为基础的带有随机搜索方式的方法，具有经过不断演化而获得逼近真实帕累托前沿的能力，因此被广泛应用到多目标问题的求解上，如NSGA-II [错误!未找到引用源。]、多目标差分(DE) [错误!未找到引用源。]、多目标粒子群(MOPSO)算法[错误!未找到引用源。]。在以种群为基础的算法中，粒子群算法具有快速收敛的特点[错误!未找到引用源。]。对于多目标粒子群优化算法，平衡收敛性和多样性是至关重要的。因此提出了很多改进策略：随机选择策略，从外部存储中随机选择一个全局最优 *gbest* 是简单且高效的策略；超体积贡献，拥挤距离等也可以作为选择 *gbest* 的指标；一个种群分不同偏好的子种群[错误!未找到引用源。]，一个负责全局搜索，另一个负责局部收敛；文献**错误!未找到引用源。**中粒子不同维度向不同的个体学习，可以增加种群多样性。EPSO [错误!未找到引用源。]引入了均衡因子，以提高种群全局搜索能力。AGE-MOPSO [错误!未找到引用源。]利用种群收敛状态，自适应

收稿日期：2018-01-24；**修回日期：**2018-04-02　**基金项目：**国家自然科学基金资助项目（61473266，61673404）；河南省高校优秀青年教师研究奖励基金资助项目（2014GGJS-004）；河南省大学创新人才科技计划资助项目（16HASTIT041）

作者简介：梁静（1981-），女，河南兰考人，教授，博士，主要研究方向为计算智能的理论研究及应用；郭倩倩（1992-），女，河南商丘人，硕士研究生，主要研究方向为智能计算、粒子群优化；岳彩通（1990-），男，博士研究生，主要研究方向为多目标优化、稀疏优化；瞿博阳（1984-），男，河南焦作人，副教授，博士，主要研究方向为智能优化算法、多目标优化（qby1984@hotmail.com）.

调整惯性权重和学习因子的值，以达到探索和开发的最佳平衡。但这些策略很少利用到多目标优化问题的特性，即在一定的条件下，*m* 个目标的多目标问题在目标空间的帕累托前沿及决策空间的帕累托解集会形成 *m*-1 维的片段连续的流型形状[9]。基于多目标问题的这种规则性，文献**错误!未找到引用源。**提出将自组织映射网络应用于种群的重组操作中，有效地提高了多目标算法的性能。但目前尚未有人将自组织映射网络与多目标粒子群优化算法结合。本文提出的自组织引导机制的自组织多目标粒子群优化算法(self-organizing multi-objective particle swarm optimization algorithm，SMPSO)，利用自组织映射网络可以将高维信息映射到低维网络，且保持特征分布不变性的特点，映射当前种群和帕累托解集的结构信息，为粒子构建新的邻域关系，引导粒子的搜索过程，以此平衡多目标粒子群算法的收敛性和多样性。

1　相关背景知识

1.1　多目标优化问题

一个连续的多目标优化问题可以表述为如下形式:

$$\begin{aligned}&\text{Min } \mathrm{F}(x)=(f_1(x), f_2(x),\dots, f_m(x))\\&\text{s.t. } x=(x_1,x_2,\dots,x_n)\in\Omega\end{aligned} \qquad (1)$$

其中：$\Omega=[a_i,b_i]^n$ 是搜索空间的可行域；x 是决策空间 Ω 中的 n 维决策变量；m 是待优化目标的个数。函数 $\mathrm{F}:\Omega\to R^m$ 定义了 m 个待优化目标函数由决策空间 Ω 到目标空间 R^m 的映射。

一些多目标问题的相关定义如下：

定义 1　帕累托支配性。决策变量 x 支配决策变量 y(记为 $x\prec y$)当且仅当满足以下条件：

$$\begin{aligned}&(\forall i\in\{1,2,\dots,m\}: f_i(x)\le f_i(y))\\&\wedge(\exists j\in\{1,2,\dots,m\}: f_i(x)< f_i(y))\end{aligned} \qquad (2)$$

即在 m 个目标函数中，x 的每个目标函数值都不大于 y 的目标函数值，且 x 至少有一个目标函数值小于 y 的目标函数值。

定义 2　帕累托解。决策变量 x 是帕累托解：

$$\neg\exists y\in\Omega: y\prec x \qquad (3)$$

即在可行域空间内，不存在任何一个粒子支配粒子 x。

定义 3　帕累托解集（Pareto optimal set，**PS**）定义为

$$\mathbf{PS}=\{x\in\Omega|\ \neg\exists y\in\Omega: y\prec x\} \qquad (4)$$

定义 4　帕累托前沿（Pareto front，**PF**）定义为

$$\mathbf{PF}=\{\mathrm{F}(x)\,|\,x\in\mathbf{PS}\} \qquad (5)$$

1.2　多目标问题的规则性

在一定条件下，一个连续的 *m* 个目标问题的 PF 和 PS 会形成 *m*-1 维的流型，如 0 所示。

鉴于这种规则性，有很多构造分布模型的算法**错误!未找到引用源。**，例如用主成分分析法构造模型**错误!未找到引用源。**，利用模型产生新解，但这种构造模型的方法，效果依赖于模型参数的选取。

1.3　粒子群优化算法

粒子群优化算法是基于种群的随机搜索的全局寻优进化算法**错误!未找到引用源。**。它通过模仿鸟类运动规则在搜索空间中寻找最优位置。与其他进化算法相似，搜索由种群个体来组成，这些搜索个体叫做粒子。粒子由两个特征即位置和速度信息组成。假设当前种群由 N 个粒子组成，粒子的搜索范围是 n 维超空间。粒子 $i\in(1,2,\dots N)$ 的位置和速度信息分别表示为 $x_i=(x_{i1},x_{i2},\dots,x_{in})$ 和 $v_i=(v_{i1},v_{i2},\dots,v_{in})$ 。其更新公式如下：

$$v_i(t+1)=w*v_i(t)+c_1r_1(x_{pbest_i}-x_i(t))+c_2r_2(x_{gbest_i}-x_i(t)) \qquad (6)$$

$$x_i(t+1)=x_i(t)+v_i(t+1) \qquad (7)$$

其中：t 是迭代代数；w 是可以用来平衡局部和全局搜索的惯性因子，较高的惯性因子给先前经验大的权值而有益于全局搜索，反之，偏向于局部搜索；c_1, c_2 代表两个加速因子影响种群的收敛速度；r_1, r_2 是在(0,1)正态分布之间产生的随机量；*pbest* 和 *gbest* 分别是个体最优和全局最优解。

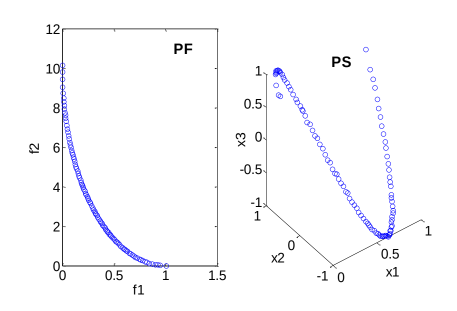


图 1　多目标问题的规则性

多目标粒子群算法中，寻找的不是唯一的 *gbest*，而是用邻域最优的个体 *nbest* 代替 *gbest*。式（6）改为如下形式：

$$v_i(t+1)=w*v_i(t)+c_1r_1(x_{pbest_i}-x_i(t))+c_2r_2(x_{nbest_i}-x_i(t)) \qquad (8)$$

2　自组织多目标粒子群优化算法 SMPSO

本文提出的 SMPSO 算法，主要是利用自组织映射网络（self-organizing mapping network, SOM）发现种群和多目标问题帕累托前沿的结构信息，然后引导粒子的飞行。同时结合了精英学习策略和多项式变异，增加种群多样性。SMPSO 流程如 0 所示。

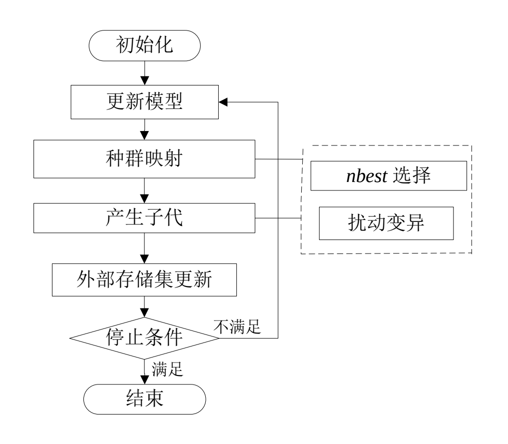


图 2　SMPSO 的流程

2.1　SOM 模型为 PSO 粒子构建邻域信息

神经网络与进化算法的结合大多是用进化算法优化神经网

络的参数错误!未找到引用源。，本文中利用自组中映射网络为粒子群算法中的粒子构建邻域关系，从而引导粒子的飞行。

如 0 所示，SOM 包括一个输入层和一个表示层。假设输入数据是 n 维，表示层有 $N = N_1 \times N_2$ 个神经元。每个神经元 $u \in (1,2,...,N)$ 赋予位置 $z^u = (z_1^u, z_2^u)$ 和权值 $w^u = (w_1^u, w_2^u, ..., w_n^u)$ 。

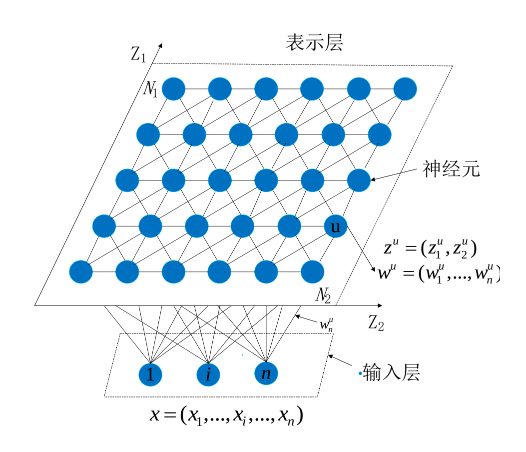


图 3　SOM 的拓扑结构示意图

SOM 的流程分为两个阶段：a)学习阶段，随机选择训练数据，根据欧氏距离选择获胜神经元，更新获胜神经元及其邻域神经元的权值；b)聚类阶段，将测试数据映射到神经元，相似的数据会映射到相邻神经元。

构建的 SOM 模型，神经元的权值 $w^u = (w_1^u, w_2^u, ..., w_n^u)$ 的初始值与种群粒子的位置信息相同，每迭代一代更新一次。进化过程中得到的新的非支配解的位置信息是 SOM 的训练数据（**TS**），SOM 更新之后，将种群（**POP**）和外部存储集（**EXA**）的粒子位置映射在网络结构上，根据神经元的位置信息，构建粒子的邻域，从而选择合适的引导粒子 *nbest*。具体步骤如算法 1 所示。

2.2　算法 1　nbest 的选择

输入：**POP**，**EXA**，**TS**，　权值 $w^u = (w_1^u, w_2^u, ..., w_n^u)$,学习率 η_0 ,学习半径 σ_0 。

输出：　$w^u = (w_1^u, w_2^u, ..., w_n^u)$ ，每个粒子的 **nbest_i**。

a)计算每个神经元之间的距离 $\left\| z^u - z^{u'} \right\|_2$ ，根据距离排序，神经元 u 的邻域神经元集合 I^u ， I_k^u 是距离神经元 u 第 k^{th} 近的神经元。

b)选择训练数据 $x = (x_1, ...x_i, ..x_n) \in \mathbf{TS}$ ，判断获胜神经元：

$$u' = \arg\min_{1 \le u \le N} \left\| x - w^u \right\|_2 \quad (9)$$

其中： $\|_\|$ 是欧氏距离。

c)更新获胜神经元及其邻域神经元 $(u \in \mathbf{U})$ 的权值：

$$\begin{aligned} &\mathbf{U} = \{u \mid 1 < u \le N \wedge \left\| z^u - z^{u'} \right\|_2 < \sigma(t)\} \\ &w_{t+1}^u = w_t^u + \eta(t) * \exp(-\left\| z^u - z^{u'} \right\|_2)(x - w_t^u) \end{aligned} \quad (10)$$

其中： $\eta(t) = \eta_0 * (1 - \frac{t}{MAXgen})$ ； $\sigma(t) = \sigma_0 * (1 - \frac{t}{MAXgen})$ ； $MAXgen$ 是最大迭代代数。

d)将待分类数据 POP、EXA 输入，根据式（9）判断获胜神经元。ind2unit 记录 POP 每个粒子的获胜神经元，unit2nbest 记录每个神经元上对应的 nbest。

e)nbest_i 记录粒子 x 的 nbest 备选库

for　i=1:NP

u= ind2unit(i); /粒子 i 的获胜神经元 u

nbest_i= unit2nbest{I_k^u};/神经元 u 的邻域神经元上对应的 nbest /

end

如 0 所示，当前粒子 i 的获胜神经元是 u，根据神经元的位置确定邻域神经元，邻域神经元上对应的 **EXA** 中的个体则组成 *nbest* 的备选库 **nbest_i**。参照快速排序法，根据占优性和拥挤距离，选择排序靠前且稀疏的个体为粒子 i 的 *nbest*。相较于将目标空间划分成几部分的分解策略错误!未找到引用源。错误!未找到引用源。，此策略有以下优点：

a)SMPSO 选出来的粒子在决策空间距离粒子较近，产生有效子代的效率更高。

b)SMPSO 在决策空间为粒子构建邻域，整个种群有不同的引导粒子，保持了种群的多样性。

c)SOM 的结构是根据目前得到的非支配解集不断学习的，权值信息不断更改，更靠近 PF，产生的解更符合问题的结构。

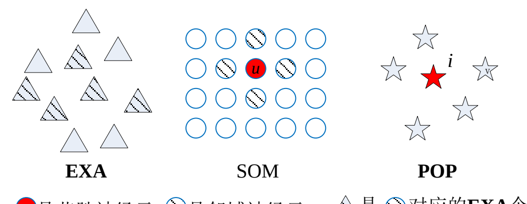


图 4　*nbest* 的选择

2.3　扰动变异策略

对于多目标粒子群算法，保留非支配解是至关重要的。因为它们是当前搜索到的最佳位置，具有一定的引导意义。但由于缺乏对待优化问题的先验知识，当种群处于收敛停滞状态时，有可能是陷入了局部最优。本文中除了使用外部存储集之外还引入了精英学习策略，在目前得到的最优位置处进行变异，有利于全局最优解的进一步寻找。

在为粒子 x 选择 *nbest* 之后，进行子代更新，更新之后的粒子按照一定概率进行精英学习。

j=rand(1,n)<pr;

$$x(j) = nbest(j) + \mathrm{normrnd}(0, pr^2) \times (b_j - a_j) \quad (11)$$

其中：a_j和b_j 是粒子 x 第 j^{th} 维的上下界；pr 的值随着迭代次数的增加 ，从 0.2 线性减少为 0.05。在进化初期，较大的

变异范围有利于算法的全局搜索，随着迭代的进行，算法需要在局部范围找到更精确的解，因此小的变异范围更适合。正态分布的概率密度如图 5 所示。由 0 可知，当 pr 值较大时，产生的随机值较大的概率比较大，从而粒子的变异范围较大，有利于产生较大的扰动；随着迭代的进行，pr 值减小，产生的随机数在均值 0 附近的概率会越来越大，变异范围减小，此时的 nbest 已接近于真实的最优位置，小范围的变异更有利于寻找精确解。

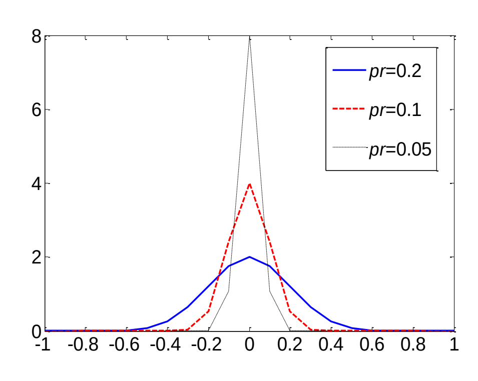


图 5　正态分布的概率密度图

同时，为了避免种群陷入局部最优，引入了多项式变异。

if　rand()<1/n

$$x(j)=x(j)+\zeta_j\times(b_j-a_j) \tag{12}$$

$$\zeta_j=\begin{cases}[2\times rnd+(1-2\times rnd\times(\frac{b_j-x(j)}{b_j-a_j})^{eta+1})]^{\frac{1}{eta+1}}-1 & \text{if } rnd<0.5\\ 1-[2-2\times rnd+(2\times rnd-1)(\frac{x(j)-b_j}{b_j-a_j})^{eta+1}]^{\frac{1}{eta+1}} & \text{otherwise}\end{cases} \tag{13}$$

其中：rnd 是在（0,1）间的随机数；eta 是变异分布指数。

2.4　SMPSO 算法的具体步骤

在 SMPSO 中，利用 SOM 模型为粒子构建邻域，SOM 权值随着种群每迭代一次进行一次更新；训练数据只采用新加入外部存储集 **EXA** 的粒子，节约计算量且避免数据训练过度。在子代更新中加入了扰动策略，平衡了收敛与多样性。对粒子位置进行了边界处理，边界粒子速度置反可以避免粒子集中于边界位置。算法 2 给出了 SMPSO 算法的具体步骤。

2.5　算法 2　SMPSO 的具体步骤

输入：种群大小 NP，PSO 参数 w, c_1, c_2。

输出：外部存储集 **EXA**。

a)初始化种群和 SOM 网络权值。

初始化种群 NP 个粒子 $\textbf{POP}=\{x^1,x^2,...,x^{NP}\}$，EXA=POP；SOM 的训练数据 TS=POP，$\{w^1,w^2,...,w^{NP}\}$=**POP**，学习率 η_0，学习半径 σ_0。

b)根据神经元之间的距离 $\|z^u-z^{u'}\|_2$，计算神经元 u 的邻域神经元集合 I^u，I_k^u 是距离神经元 u 第 k^{th} 近的神经元。

c)循环。

(a)SOM 权值更新：根据 TS 及式（9）（10）更新网络权值。

(b)参考算法 1，为粒子选择 nbest 备选库，按照 NSGA-II 的快速排序法对备选库的粒子进行排序，选择排序靠前且稀疏的粒子作为 nbest。

(c)子代更新。

在 SMPSO 中，粒子群更新公式为

$$v=w\times v+rand\times c_1\times(pbest-x)+rand\times c_2\times(nbest-x) \tag{14}$$

其中：$w=w_{\max}-(w_{\max}-w_{\min})*\frac{t}{MAXgen}$。

速度边界处理：超过边界的值赋予边界值。

$$x=x+v \tag{15}$$

之后，按照一定概率执行精英学习策略，参考式（11）。

位置边界处理：超过边界的值赋予边界值且速度置反。

进行多项式变异操作，参考式（12）和（13）。

(d)更新 EXA 和 TS。

外部存储集 EXA 的更新策略，根据占优性和拥挤距离排序，保留排序靠前且稀疏的粒子；

训练数据 TS=unique（EXA,TS），即只把新产生的加入 EXA 的粒子作为训练数据。

d)结束，否则返回步骤 c)。

3　仿真实验与分析

3.1　实验设置

3.1.1 测试函数

本文采用文献**错误!未找到引用源。**中的 GLT1-GLT6 和文献**错误!未找到引用源。**中 DLTZ2、DLTZ4 和 DLTZ5 为测试函数。0 列出了测试函数的决策变量和 Pareto 前沿的信息。

表 1　用于算法性能比较的多目标优化测试问题

测试问题	目标维数	变量维数	变量范围	Pareto 前沿特点
GLT1	2	10	$[0,1]\times[-1,1]^{n-1}$	线性 不连续
GLT2	2	10	$[0,1]\times[-1,1]^{n-1}$	凸 PF 连续
GLT3	2	10	$[0,1]\times[-1,1]^{n-1}$	凸 PF 连续
GLT4	2	10	$[0,1]\times[-1,1]^{n-1}$	凸 PF 不连续
GLT5	3	10	$[0,1]^2\times[-1,1]^{n-2}$	凸 PF 连续
GLT6	3	10	$[0,1]^2\times[-1,1]^{n-2}$	凸 PF 不连续
DTLZ2	3	12	$[0,1]^n$	凹形，连续
DTLZ4	3	12	$[0,1]^n$	凹形 非均匀
DTLZ5	3	12	$[0,1]^n$	线型 连续

3.1.2 比较算法

为了比较算法(SMPSO)的性能，选取三种代表性的多目标优化的对比算法。这些算法中有 NSGAII **错误!未找到引用源。**、2LBMOPSO **错误!未找到引用源。**、pccsAMOPSO **错误!未找到引用源。**、IM-MOEA **错误!未找到引用源。** 和 MOEA/IGD-NS **错误!未找到引用源。**。各算法的实验参数都按照对应参考文献设置。

3.1.3 实验参数

实验中 GLT1-GLT4 是 2 目标的多目标问题，种群大小设为 100，SOM 的网络结构设置为一维（1×100），外部存储容量

均为 100，测试函数的最大评价次数为 50 000。GLT5-GLT6 和 DTLZ 系列是 3 目标的多目标问题，种群大小设为 150，SOM 的网络结构设置为二维（15×10），外部存储容量均为 150，测试函数的最大评价次数为 70 000。 每种算法在所有测试函数上均独立运行 25 次。

3.1.4 性能指标

为了评估算法获得的近似 PF 的收敛性和均匀性，本文采用反转世代距离(inverted generational distance，IGD)和超体积值(hyper-volume，HV)作为性能评估指标。令 $\mathbf{P}$ 为真实 Pareto 最优解集，$\mathbf{P}^*$ 是得到的 Pareto 最优解集，则 *IGD* 的计算如下：

$$IGD(\mathbf{P}^*,\mathbf{P})=\frac{\sum_{x^*\in\mathbf{P}^*}d(x^*,\mathbf{P})}{\left|\mathbf{P}^*\right|} \tag{16}$$

其中：$d(x^*,\mathbf{P})$ 是 x^* 到 $\mathbf{P}$ 中任何一个点的最小距离；$\left|\mathbf{P}^*\right|$ 是 $\mathbf{P}^*$ 的基数。

HV 的计算方式为

$$HV(\mathbf{P},\boldsymbol{r})=VOL(\bigcup_{x\in\mathbf{P}}[f_1(x),r_1]\times\ldots[f_m(x),r_m]) \tag{17}$$

其中：$\boldsymbol{r}=(r_1,\ldots,r_m)$ 是目标空间里被任何帕累托前沿占优的参考点；$VOL(.)$ 是勒贝格测度。GLT1、GLT3 的参考点 $\boldsymbol{r}=(2,2)$，GLT2 的 $\boldsymbol{r}=(2,11)$，GLT4 的参考点 $\boldsymbol{r}=(2,3)$，GLT5-GLT6、DLZ2 和 DLZ4-DLZ5 的参考点 $\boldsymbol{r}=(2,2,2)$。

实验中 *IGD* 和 *HV* 的值都用来衡量种群的收敛性和分布性，*IGD* 值越小（*HV* 的值越大），则得到的 PF 越好。得到的 *IGD* 值很小且 *HV* 值很大，则得到 PF 非常接近真实的 PF，且能完全覆盖真正的 PF。

3.2　实验结果及分析

SMPSO 算法中需要更新拓扑结构的权值，且在一定概率下进行变异。为了比较各种算法的时间复杂度，将 6 个比较算法在 9 个多目标测试问题上独立运行 25 次的耗时均值列于 0 内。可以看出 SMPSO 算法的时间复杂度比 NSGAII 和 IM-MOEA 的大，但保持在同一数量级，在 3 目标的测试问题上耗时比 2LBMOPSO，pccsAMOPSO 和 MOEA/IGD-NS 相对较少。

0 和 0 列出了 6 种比较算法在 9 个多目标测试问题上的 *IGD* 和 *HV* 性能指标值。其中，平均值(*mean*)和标准差值(*std*)是同一算法在同一测试问题上独立运行 25 次的统计结果；同时，采用粗体字和灰色背景表示所有对比算法在每一个测试问题中的最小 *IGD* 值和最大 *HV* 值。每种算法对同一个问题的排名及综合排名列在表内。同时用秩和检验，5%的显著性水平比较各算法与 SMPSO 的显著性差异。分别用“†”、“§”和“≈”表示算法 SMPSO 比对比算法较好，较差和没有明显差异的情况。

表 2　各种算法在每个测试问题上的平均耗时/s

	2LBMOPSO	NSGAII	pccsAMOPSO	IM-MOEA	MOEA/IGD-NS	SMPSO
GLT1	62.14	12.38	82.72	18.49	5.49	46.3
GLT2	63.75	10.86	98.07	19.57	5.94	45.35
GLT3	68.88	9.97	143.67	24.22	24.38	45.36
GLT4	72.42	12.03	84.86	20.41	10.78	50.41
GLT5	138.1	19.91	138.20	25.68	183.45	76.11
GLT6	140.84	20.99	602.35	23.66	176.82	74.56
DTLZ2	134.24	20.16	635.679	20.12	426.54	76.41
DTLZ4	131.33	20.7	543.43	20.45	372.61	79.87
DTLZ5	130.42	20.38	605.28	22.92	390.28	78.56

表 3　测试函数性能度量 *IGD* 的均值(标准差)

函数		2LBMOPSO	NSGAII	pccsAMOPSO	IM-MOEA	MOEA/IGD-NS	SMPSO
GLT1	*mean*	1.17E-01†[6]	4.35E-02$^{\approx}$[3]	8.12E-02†[4]	3.61E-02$^{\approx}$[2]	9.88E-02†[5]	**2.38E-02[1]**
	std	4.40E-02	3.07E-02	2.41E-02	1.95E-02	5.22E-03	2.25E-02
GLT2	*mean*	1.12E-01†[3]	4.28E-02†[2]	7.12E-01†[5]	6.59E-01†[4]	1.95E+00†[6]	**3.82E-02[1]**
	std	2.20E-01	2.89E-03	4.67E-01	2.22E-01	1.60E-01	1.88E-03
GLT3	*mean*	6.64E-02†[5]	2.81E-02†[2]	1.12E-01†[6]	4.17E-02†[3]	6.53E-02†[4]	**9.32E-03[1]**
	std	2.17E-02	1.46E-02	4.46E-02	8.31E-03	5.53E-03	5.32E-03
GLT4	*mean*	1.32E-01†[4]	1.16E-01†[3]	2.27E-01†[5]	6.14E-02†[2]	2.62E-01†[6]	**4.17E-02[1]**
	std	7.04E-02	7.80E-02	9.81E-02	4.16E-02	1.50E-02	4.90E-02
GLT5	*mean*	5.14E-02†[4]	6.51E-02†[5]	8.88E-02†[6]	4.86E-02$^{\approx}$[3]	**3.43E-02§[1]**	4.77E-02[2]
	std	2.70E-03	4.57E-03	1.84E-02	2.96E-03	1.25E-03	2.57E-03

GLT6	*mean*	4.99E-02†[4]	5.83E-02†[5]	7.34E-02†[6]	3.18E-02≈[2]	**2.71E-02≈[1]**	4.12E-02[3]
	std	1.77E-03	5.37E-03	3.59E-02	1.26E-03	1.56E-03	3.34E-03
DTLZ2	*mean*	6.15E-02†[4]	1.00E-01†[6]	5.27E-02§[2]	7.78E-02†[5]	**4.29E-02§[1]**	5.51E-02[3]
	std	2.46E-03	7.48E-03	1.69E-03	3.50E-03	3.65E-04	2.04E-03
DTLZ4	*mean*	6.28E-02†[2]	3.15E-01† [6]	1.05E-01†[4]	6.73E-02† [3]	2.12E-01†[5]	**5.40E-02[1]**
	std	1.64E-03	7.99E-02	1.16E-01	2.38E-03	3.02E-01	2.05E-03
DTLZ5	*mean*	4.36E-03†[5]	3.69E-03≈[4]	3.37E-03§[2]	1.57E-02†[6]	**2.80E-03§[1]**	3.68E-03[3]
	std	2.68E-04	5.14E-04	2.16E-04	1.96E-03	2.19E-04	2.41E-04
Rank		4.111	4.000	4.444	3.333	3.333	1.778
†/ § /≈		9/0/0	7/0/2	7/0/2	6/0/3	5/3/1	

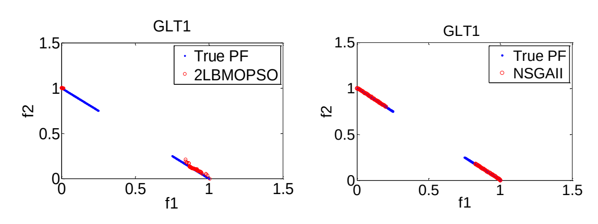
表 4　测试函数性能度量 *HV* 的均值(标准差)

函数		2LBMOPSO	NSGAII	pccsAMOPSO	IM-MOEA	MOEA/IGD-NS	SMPSO
GLT1	*mean*	～～[6]	3.26E+00≈[3]	3.17E+00†[4]	3.28E+00≈[2]	3.17E+00†[5]	**3.31E+00[1]**
	std	～～	6.84E-02	5.38E-02	3.44E-02	2.42E-02	5.30E-02
GLT2	*mean*	1.84E+01†[4]	1.97E+01†[2]	1.83E+01†[5]	1.90E+01†[3]	1.64E+01†[6]	**1.98E+01[1]**
	std	1.11E-02	7.03E-03	6.47E-01	2.34E-01	5.51E-01	1.01E-02
GLT3	*mean*	3.93E+00†[5]	3.94E+00†[4]	3.91E+00†[6]	3.94E+00†[3]	3.94E+00≈[2]	**3.95E+00[1]**
	std	4.03E-03	2.75E-03	3.70E-02	1.74E-03	1.13E-03	1.06E-03
GLT4	*mean*	4.86E+00†[2]	4.85E+00†[3]	4.40E+00†[6]	4.81E+00†[4]	4.42E+00†[5]	**4.97E+00[1]**
	std	1.27E-01	1.83E-01	9.14E-01	1.35E-01	2.07E-02	2.59E-02
GLT5	*mean*	3.88E+00†[3]	3.87E+00†[4]	3.85E+00†[5]	3.81E+00†[6]	3.88E+00†[2]	**3.90E+00[1]**
	std	1.31E-02	2.35E-02	2.75E-02	2.03E-02	1.56E-02	1.56E-02
GLT6	*mean*	3.85E+00†[4]	3.84E+00†[6]	3.86E+00≈[2]	3.85E+00†[5]	**3.89E+00§[1]**	3.86E+00[3]
	std	1.08E-02	1.42E-02	3.32E-02	2.34E-02	1.27E-02	1.37E-02
DTLZ2	*mean*	3.48E+00†[4]	3.47E+00†[5]	3.49E+00≈[2]	3.48E+00†[3]	3.46E+00†[6]	**3.49E+00[1]**
	std	2.46E-03	7.48E-03	1.69E-03	4.25E-02	2.94E-02	3.59E-03
DTLZ4	*mean*	3.48E+00†[5]	**3.57E+00§**[1]	3.55E+00§[2]	3.52E+00§[3]	3.47E+00†[6]	3.50E+00[4]
	std	1.64E-03	7.99E-02	1.16E-01	2.63E-02	2.08E-01	2.05E-03
DTLZ5	*mean*	**2.84E+00≈[1]**	2.84E+00†[5]	2.84E+00≈[2]	2.84E+00†[4]	2.83E+00†[6]	2.84E+00[3]
	std	2.68E-04	5.14E-04	2.16E-04	2.30E-02	6.19E-03	2.16E-04
Rank		3.778	3.667	3.778	3.667	4.333	1.778
†/ § /≈		8/0/1	7/1/1	5/1/3	7/1/1	7/1/1	

从 0 中可以看出，SMPSO 算法对大多测试问题都有很好的性能度量。对于测试问题 GLT1-GLT4 和 DTLZ4，SMPSO 算法得到的 *IGD* 均值最小。对于测试问题 GLT5，SMPSO 的 *IGD* 值仅大于 MOEA/IGD-NS 算法，对于 GLT6、DLTZ2 和 DLTZ5，SMPSO 算法的 *IGD* 指标排名第三。表中的 *Rank* 值是算法对所有测试问题的平均排名，可以看出 SMPSO 的排名最高。且秩和检验的结果表明，SMPSO 在大多测试问题上明显优于其他对比算法。

0 中“～～”表示 2LBMOPSO 算法对测试问题 GLT1 的结果较差，性能指标 *HV* 出现负值，不作统计。SMPSO 算法对于 GLT1-GLT5 和 DTLZ2，*HV* 值均为最大。对于 GLT6，DLTZ5 测试问题，*HV* 均值排名第三。当然，SMPSO 不可能在所有测试问题上效果都比较好，如在 DTLZ4 上效果较差。综合排名和显著性水平显示，SMPSO 对大部分测试问题都有较好的多样性。

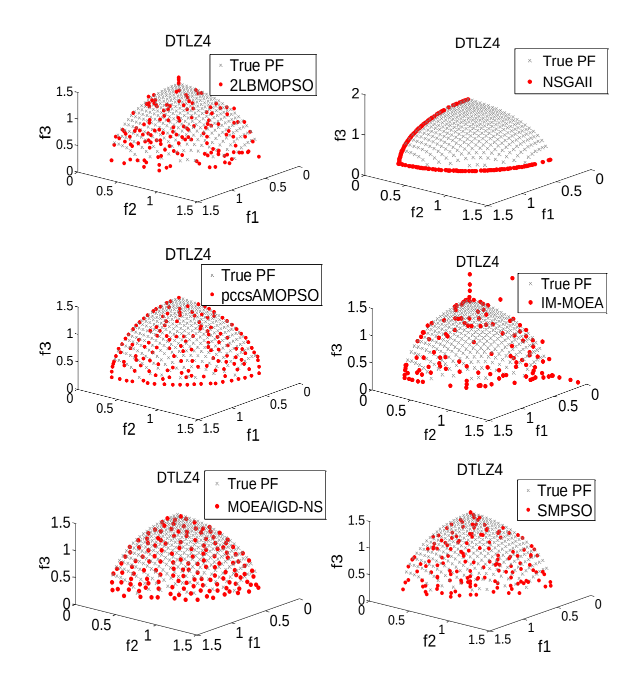
为了进一步显示各种算法求得最优解集的收敛和多样性的情况，以测试问题 GLT1、GLT3、GLT6、DTLZ4-DTLZ5 为例，将测试函数运行 25 次结果，按 *IGD* 值排名中间的一次运行结果显示如 0 所示。



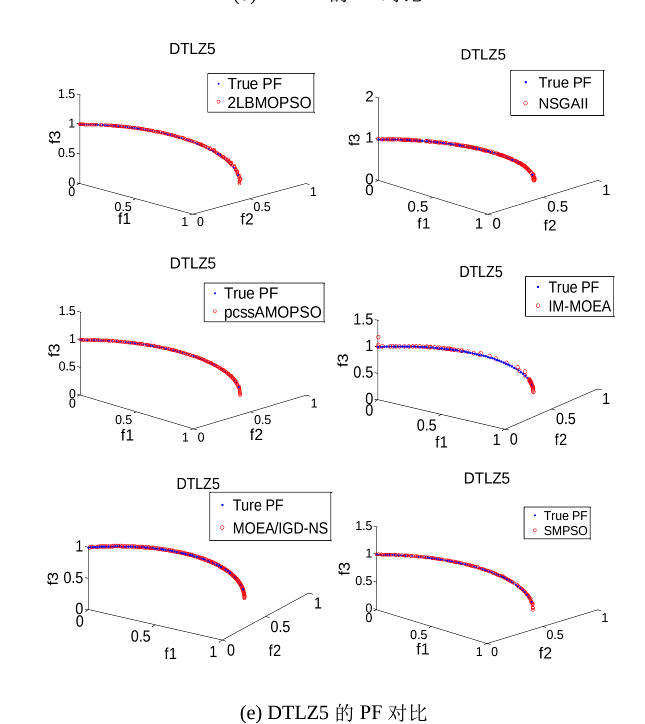
(a) GLT1 的 PF 对比

(b) GLT3 的 PF 对比

(c) GLT6 的 PF 对比



(d) DTLZ4 的 PF 对比



(e) DTLZ5 的 PF 对比

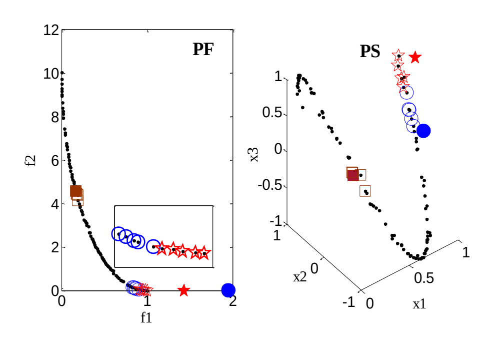
图 6 测试函数 PF 对比

从 0 中可以看出，无论是 2 目标还是 3 目标问题，SMPSO 得到的 PF 均可以较好地覆盖真正的 PF，相对于对比算法，SMPSO 能较好的处理 PF 不连续的情况，如 GLT1 和 GLT6，在 SMPSO 中的 SOM 网络结构可以较好地发现问题的结构信息，且网络结构的整体性有利于种群的全局搜索，因此找到的 PF 没有缺失。GLT3 虽然 PF 是连续的，但是测试函数的复杂，导致找到的 PF 能完全覆盖真正的帕累托前沿有一定的难度。DTLZ4 的问题上，SMPSO 得到的 PF 虽然能较全面地覆盖真正的 PF，但图中显示分布的均匀性略差。DTLZ5 测试问题比较简单，大部分算法得到的帕累托前沿能够靠近真正的 PF。

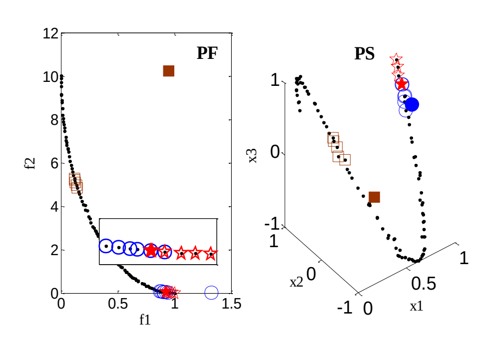
为了验证用 SOM 构造邻域关系的合理性，0 将以 GLT2 问

题为例，列出在进化过程中神经元上对应的粒子及粒子的 *nbest* 备选库在决策空间和目标空间的分布。

从图中可以看出，在进化的过程中，为粒子选择的 *nbest* 不论在 PF 或是 PS 空间，都是外部存储集 **EXA** 中距离粒子个体比较近的，有利于局部搜索，增加产生高质量粒子的可能性。粒子 a 和 b 对应的神经元在拓扑结构中距离较近，它们映射的粒子距离也比较近。同理，c 粒子对应的神经元在拓扑结构上，距离较远，则 a、b 距离 c 粒子及其邻域较远。这也是 SOM 的特性之一，即相似的个体聚在一起。每进化一代之后，权值学习一次，因此权值的调整也是慢慢累积的。图(b)中粒子 c 在 PF 空间分布距离邻域较远，是因为 SOM 的映射关系是在 PS 空间建立的，所以反映在 PF 空间会有所偏差。



(b)　第 200 代时粒子的分布



(b)第 400 代时粒子的分布

. 是当前外部存储集 **EXA**；

★是粒子 a，第 40 号神经元对应的粒子，☆是粒子 a 的 *nbest* 备选；

●是粒子 b，第 45 号神经元对应的粒子，○是粒子 b 的 *nbest* 备选；

■是粒子 c，第 80 号神经元对应的粒子，□是粒子 c 的 *nbest* 备选。

图 7　不同粒子的邻域分布

SMPSO 算法中精英变异策略中的参数 *pr* 初始值的选取，在参考了数学模型的概率分布之外，另外进行了实验的验证。如 0 所示，设定 *pr* 最小值为 0.05，*pr* 初始值分别设为 0.1、02、0.3、0.4 和 0.5，折线图显示没有某一个 *pr* 初始值可以使所有测试问题的 *IGD* 值均为最小，但是 *pr* 初始值设为 0.2 时可以使大多测试问题的 *IGD* 值较小，且变异概率较小可以降低计算复杂度。因此 *pr* 初始值设为 0.2，*pr* 线性减小到最小值为 0.05 而不是 0，是为了在算法的后期仍能产生一定的扰动。

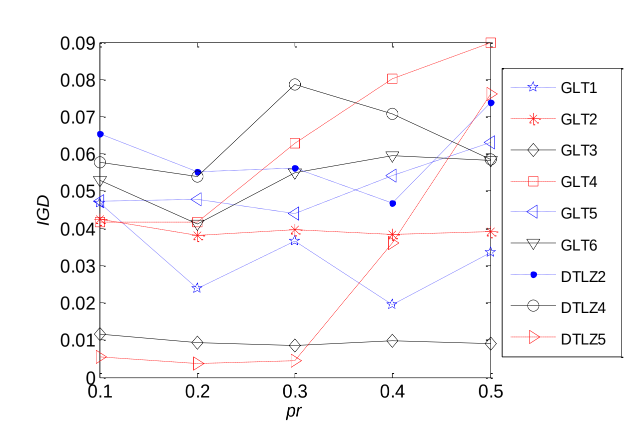
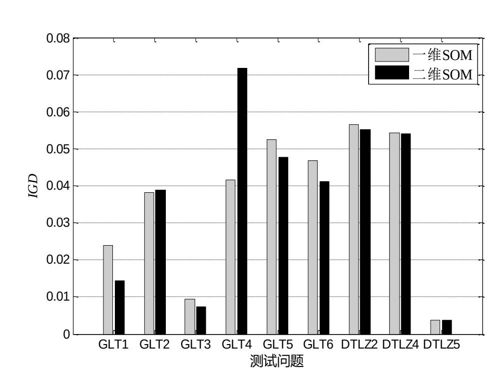
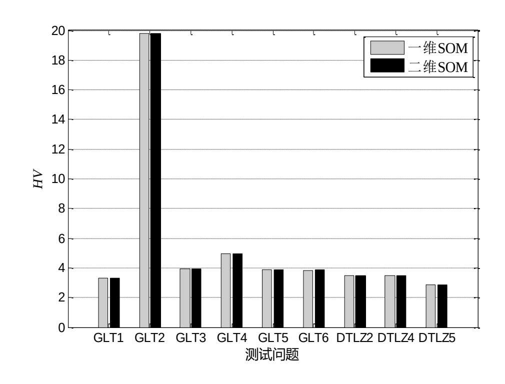


图 8　不同 *pr* 初始值对各个测试问题 *IGD* 值的影响

SMPSO 中根据多目标优化问题的规则性，2 目标问题，采用一维（1×100）SOM，3 目标问题，采用二维（15×10）SOM。为了验证 SOM 拓扑结构对算法性能指标 *IGD* 和 *HV* 的影响，将 SOM 的拓扑结构作出如下改变：2 目标问题改用二维（10×10）SOM，3 目标问题改用一维（1×150）SOM，将算法独立运行 25 次的平均值结果显示如 0 所示。



(a) SOM 拓扑结构对 *IGD* 的影响



(b) SOM 拓扑结构对 *HV* 的影响

图 9　SOM 拓扑结构对 *IGD* 和 *HV* 的影响

从图 9（a）中可以看出，对于 2 目标的测试问题 GLT1、GLT3，*IGD* 值在二维 SOM 的情况下较小，GLT2、GLT4,在一维 SOM 的情况下较小，3 目标的测试问题在二维 SOM 时 *IGD* 值较小；从（b）中可以看出，SOM 拓扑结构的改变对 2 目标问题的 *HV* 值影响不大，3 目标问题的 *HV* 值在二维 SOM 的情况下略高。因此，总的来说，2 目标问题对一维或二维拓扑结构的 SOM 无明显偏好，3 目标问题较偏好二维拓扑结构的 SOM。

4 结束语

通过结合多目标问题的规则性和自组织映射网络的映射关系不变性，本文提出了一种新算法。该算法为多目标粒子群算法构建了一种新的邻域关系，为每个粒子选取了更合适的引导粒子，同时加入了精英学习机制。算法兼顾了收敛性和多样性，对大多的多目标问题都能得到分布均匀且靠近真正 PF 的解集。在后续研究中，将该算法应用到高维多目标问题上，并进一步改进算法的性能。

参考文献：

[1] Deb K. A fast elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2000, 6 (2): 182-197.

[2] 林震，侯杏娜，韦晓虎. 基于动态多策略差分进化模型的 MOEA//D 算法 [J]. 计算机应用研究, 2017, 34 (9): 2624-2628. (Lin Zhen, Hou Xingna, Wei Xiaohu. MOEA//D based on dynamic multi-strategy differential evolution model [J]. Application Research of Computers, 2017, 34 (9): 2624-2628.)

[3] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8 (3): 256-279.

[4] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. 2002 (4): 1942-1948.

[5] 韩敏，张丽君. 基于多样性检测的双子群多目标粒子群算法 [J]. 控制与决策, 2017, 32 (12): 2268-2272. (Han Min, Zhang Lijun. Bi-group multi-objective particle swarm optimization algorithm based on diversity metric [J]. Control and Decision, 2017, 32 (12): 2268-2272.)

[6] 肖闪丽，王宇嘉，聂善坤. 动态邻居维度学习的多目标粒子群算法 [J]. 计算机工程与应用, 2017, 53 (20): 31-38. (Xiao Shanli, Wang Yujia, Nie Shankun. Multi-objective particle swarm optimization based on dynamic neighborhood for dimensional learning [J]. Computer Engineering and Applications, 2017, 53 (20): 31-38.)

[7] 夏星宇，高浩，王创业. 均衡策略粒子群算法在图像分割中的应用 [J]. 郑州大学学报：工学版, 2018, 39 (1): 59-66. (Xia Xingyu, Gao Hao, Wang Chuangye. Quantification calculation and modeling simulation of distribution network losses considering harmonic factor [J]. Journal of Zhengzhou University: Engineering Science, 2018, 39 (1): 59-66.)

[8] 韩敏，何泳. 基于高斯混沌变异和精英学习的自适应多目标粒子群算法 [J]. 控制与决策, 2016, 31 (8): 1372-1378. (Han Min, He Yong. Adaptive multi-objective particle swarm optimization with Gaussian chaotic mutation and elite learning [J]. Control and Decision, 2016, 31 (8): 1372-1378.)

[9] Schütze O, Mostaghim S, Dellnitz M, *et al.* Covering pareto sets by multilevel evolutionary subdivision techniques [C]// Proc of Evolutionary Multi-Criterion Optimization, Second International Conference. 2003: 118-132.

[10] Zhang Hu, Zhou Aimin, Song Shenmin, *et al.* A self-organizing multi-objective evolutionary algorithm [J]. IEEE Trans on Evolutionary Computation, 2016, 20 (5): 792-806.

[11] 时光，刘健辰，陈忠华，等. 基于 DE-EDA 多目标优化的受电弓模糊控制 [J]. 计算机工程与应用, 2016, 52 (1): 229-232. (Shi Guang, Liu Jianchen, Chen Zhonghua, *et al.* Control of pantograph based on DE-EDA multi-objective optimization [J]. Computer Engineering and Applications, 2016, 52 (1): 229-232.)

[12] Wang Handing, Zhang Qingfu, Jiao Licheng, *et al.* Regularity model for noisy multi-objective optimization [J]. IEEE Trans on Cybernetics, 2016, 46 (9): 1997-2009.

[13] 毛晓波，张群，梁静，等. 基于 PSO-RBF 神经网络的雾霾车牌识别算法研究 [J]. 郑州大学学报：工学版, 2017, 38 (4): 46-50. (Mao Xiaobo, Zhang Qun, Liang Jing, *et al.* The haze plate recognition system based on PSO-RBF neural network [J]. Journal of Zhengzhou University: Engineering Science, 2017, 38 (4): 46-50.)

[14] Wang Hu, Yen G G. Adaptive multi-objective particle swarm optimization based on parallel cell coordinate system [J]. IEEE Trans on Evolutionary Computation, 2015, 19 (1): 1-18.

[15] Zhao Shizheng, Suganthan P N. Two-lbests based multi-objective particle swarm optimizer [J]. Engineering Optimization, 2011, 43 (1): 1-17.

[16] Gu Fangqing, Liu Hailin, Tan K C. A multiobjective evolutionary algorithm using dynamic weight design method [J]. International Journal of Innovative Computing Information & Control, 2012, 8 (5): 3677-3688.

[17] Deb K, Thiele L, Laumanns M, *et al.* Scalable test problems for evolutionary multiobjective optimization [M]// Evolutionary Multiobjective Optimization. 2005: 105-145

[18] Cheng Ran, Jin Yaochu, Narukawa K, *et al.* A Multiobjective evolutionary algorithm using Gaussian process-based inverse modeling [J]. IEEE Trans on Evolutionary Computation, 2015, 19 (6): 838-856.

[19] Tian Ye, Zhang Xingyi, Cheng Ran, *et al.* A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric [C]// Proc of IEEE Congress on Evolutionary Computation. 2016: 5222–5229.